

AMENDMENTS TO THE CLAIMS

This listing of claim will replace all prior versions and listings of claim in the application.

1. – 92. (Cancelled)

93. (previously presented) A method for providing a user interface, comprising:
sending a request for content from an Internet browser to a server;
receiving an interface engine in response to said request, said interface engine includes a set of views, layouts, constraints and animators;
receiving requests to change appearances of items displayed in said browser that are represented by said views; and
changing said appearances of said items in said browser by providing continuous fluid transitions of appearances of said items using said layouts, constraints and animators.

94. (previously presented) A method according to claim 93, wherein:
said requests to change appearances includes a request to change a position of a first item in said user interface.

95. (previously presented) A method according to claim 93, wherein:
said requests to change appearances includes a request to change a size of a first item in said user interface.

96. (previously presented) A method according to claim 93, wherein said changing said appearances of said items includes:
calling one or more attribute modifiers; and
modifying a first view in response to said one or more attribute modifiers.

97. (previously presented) A method according to claim 96, wherein:
said calling one or more attribute modifiers includes calling a first animator to animate a property of said first view; and
said modifying a first view includes animating said property in response to said first animator.

98. (previously presented) A method according to claim 97, further comprising:
calling a constraint to modify an attribute in response to animating said property; and
modifying said attribute in response to said constraint.

99. (previously presented) A method according to claim 96, wherein:
said first view identifies a set of child views;
said calling one or more attribute modifiers includes calling a layout to layout one or more of said child views; and
said modifying a first view includes laying out said one or more child views in response to said layout.

100. (cancelled)

101. (currently amended) A method according to claim 104 ~~100~~, further comprising:
sending a request to a server via a network;
receiving an interface engine in response to said request to said server; and
running said interface engine within said network browser, said implementing said requested change is performed by said interface engine.

102. (cancelled)

103. (cancelled)

104. (currently amended) ~~A method according to claim 103, wherein:~~ A method for providing a user interface, comprising:

receiving a request to change an item displayed in a network browser from a first visual state to a second unplanned visual state, said item is associated with a first view of a set of views that comprise said user interface; and

implementing said requested change by providing continuous fluid transitions for said item from said first visual state to said second unplanned visual state, said continuous fluid transitions are performed by a method comprising calling one or more attribute modifiers for said first view and modifying said first view in response to said one or more attribute modifiers, said attribute modifiers includes layouts, constraints and animators.

105. (currently amended) A method according to claim 104 ~~100~~, wherein:
said first visual state pertains to a first position; and
said second unplanned visual state pertains to a second position.

106. (currently amended) A method according to claim 104 ~~100~~, wherein:
said first visual state pertains to a first size; and
said second unplanned visual state pertains to a second size.

107. (previously presented) An apparatus that provides a user interface, comprising:
a storage device;
a communications interface in communication with a network;
a display; and
one or more processors in communication with said storage device, said communications interface and said display;
said one or more processors implement a browser that receives an interface engine from a server in communication with said browser via said network, said interface engine implements a user

interface within said browser, said interface engine includes a set of views, layouts, constraints and animators;

said browser receives a requests to change appearances of items displayed in said browser that are represented by said views and changes said appearances by providing continuous fluid transitions of appearances of said items using said layouts, constraints and animators.

108. (previously presented) An apparatus according to claim 107, wherein:
said requests to change appearances includes a request to change a position of an item in said user interface.

109. (previously presented) An apparatus according to claim 107, wherein:
said requests to change appearances includes a request to change a size of an item in said user interface.

110. (previously presented) An apparatus according to claim 107, wherein said changing said appearances of said items includes:
calling one or more attribute modifiers, said attribute modifiers includes said layouts, constraints and animators; and
modifying a first view in response to said one or more attribute modifiers.

111. (previously presented) An apparatus according to claim 110, wherein:
said calling one or more attribute modifiers includes calling a first animator to animate a property of said first view; and
said modifying a first view includes animating said property in response to said first animator.

112. (previously presented) An apparatus according to claim 111, wherein:
said interface engine calls a constraint to modify an attribute in response to animating said

property and modifies said attribute in response to said constraint.

113. (previously presented) One or more processor readable storage devices having code embodied on said one or more processor readable storage devices, said code for programming one or more processors to perform a method comprising:

 sending a request for content from an Internet browser to a server;

 receiving an interface engine in response to said request, said interface engine includes a set of views, layouts, constraints and animators;

 receiving requests to change appearances of items displayed in said browser that are represented by said views; and

 changing said appearances of said items in said browser by providing continuous fluid transitions of appearances of said items using one or more attribute modifiers, said attribute modifiers include said layouts, constraints and animators.

114. (previously presented) One or more processor readable storage devices according to claim 113, wherein:

 said requests to change appearances includes a request to change a position of a first item in said user interface.

115. (previously presented) One or more processor readable storage devices according to claim 113, wherein:

 said requests to change appearances includes a request to change a size of a first item in said user interface.

116. (previously presented) One or more processor readable storage devices according to claim 113, wherein:

 said changing said appearances includes calling a first animator to animate a property of a first view.

117. (previously presented) One or more processor readable storage devices according to claim 113, wherein:

said changing said appearances further includes calling a first constraint to modify an attribute of a first view in response to a second view.

118. (previously presented) One or more processor readable storage devices according to claim 113, wherein:

said changing said appearances includes changing a first view;

said first view identifies a set of child views; and

said changing a first view includes calling a layout to layout one or more of said child views.

119. (cancelled)

120. (currently amended) One or more processor readable storage devices having code embodied on said one or more processor readable storage devices, said code for programming one or more processors to perform a method comprising: according to claim 119, wherein:

receiving a request to change an item displayed in a network browser from a first visual state to a second unplanned visual state; and

implementing said requested change by providing continuous fluid transitions for said item from said first visual state to said second unplanned visual state;

said code includes views, layouts, constraints and animators;

said item is associated with a first view;

said continuous fluid transitions are provided by a method comprising calling one or more attribute modifiers for said first view and modifying said first view in response to said one or more attribute modifiers; and

said one or more attribute modifiers includes said layouts, constraints and animators.

121. (currently amended) One or more processor readable storage devices having code embodied on said one or more processor readable storage devices, said code for programming one or more processors to perform a method comprising: ~~according to claim 119, wherein:~~

receiving a request to change an item displayed in a network browser from a first visual state to a second unplanned visual state; and

implementing said requested change by providing continuous fluid transitions for said item from said first visual state to said second unplanned visual state;

said code includes views, layouts, constraints and animators;

said item is associated with a first view;

said continuous fluid transitions are provided by calling one or more animators to animate said first view in response to said request to change; ~~and~~

said calling one or more animators includes informing said one or more animators of said second unplanned visual state.

122. (currently amended) One or more processor readable storage devices according to claim 120 ~~119~~, wherein:

said first visual state pertains to a first position; and

said second unplanned visual state pertains to a second position.

123. (currently amended) One or more processor readable storage devices according to claim 120 ~~119~~, wherein:

said first visual state pertains to a first size; and

said second unplanned visual state pertains to a second size.